

Grau en Matemàtiques

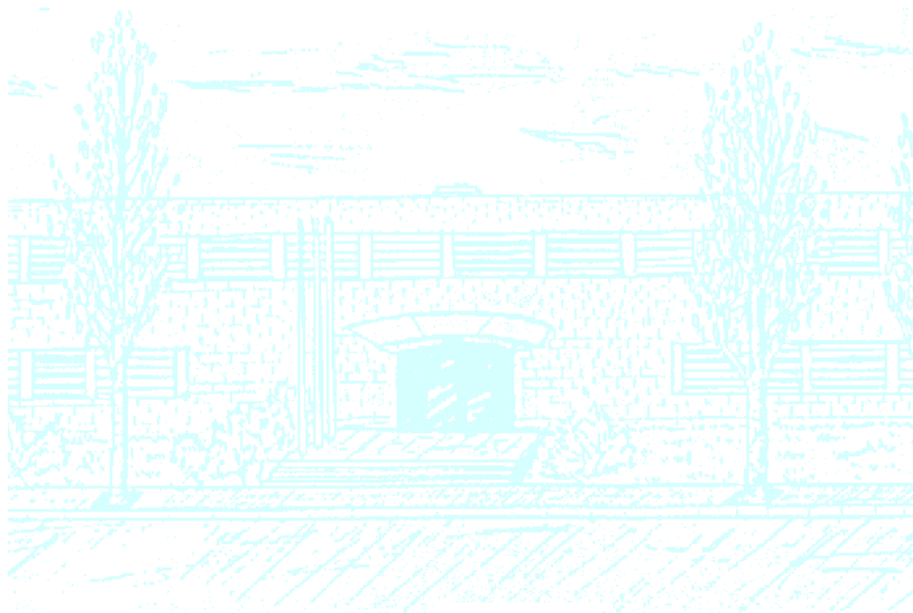
Títol: Model de Phase-Field dinàmic per a fractura fràgil

Autor: Sergi de la Torre Israel

Director: Sonia Fernández-Méndez, Alba Muixí Ballonga

Departament: Departament d'Enginyeria Civil i Ambiental

Convocatòria: 2018 - 2019



Universitat Politècnica de Catalunya
Facultat de Matemàtiques i Estadística

Grau en Matemàtiques
Treball de Fi de Grau

Model de Phase-Field dinàmic per a fractura fràgil

Sergi de la Torre Israel

Supervisat per Sonia Fernández-Méndez & Alba Muixí Ballonga

Juny 2019

En primer lloc, volia donar les gràcies a les meves tutores, la Sonia i l'Alba, per la seva dedicació i ajuda. També agrair a la meva família i amics el suport rebut al llarg del desenvolupament del treball.

Resum

Els models de phase-field per a fractura fràgil descriuen les fractures com a zones danyades mitjançant un camp continu que varia entre els estats intacte i trencat. En aquest treball, implementem numèricament un model de phase-field dinàmic i el provem resolent dos problemes de fractura clàssics. Addicionalment, fem una anàlisi del paràmetre material G_c i discutim una estratègia per adaptar malles en problemes de fractura.

Paraules clau

Phase-field dinàmic, Fractura fràgil, Branching

Índex

1	Introducció	3
1.1	Estructura i objectius del treball	3
2	Model de Phase-Field dinàmic	5
3	Test de l'L	7
4	Crack branching dinàmic	10
4.1	Test de crack branching dinàmic	10
4.2	Anàlisi de G_c	13
5	Mètode numèric	15
5.1	Esquema numèric	15
5.2	Equació d'Elasticitat Lineal Transitòria	16
5.3	Mètode de Newmark	18
5.4	Equació de dany	19
6	Creació de malles	21
6.1	EZ4U i Background Mesh	21
6.2	Determinació dels valors nodals per la background mesh	22
7	Conclusions	24
7.1	Coneixements adquirits	24
7.2	Futurs projectes	24
A	Forma feble de l'Equació d'Elasticitat Lineal Transitòria	26
B	Codis	27
B.1	Implementació del model de phase-field dinàmic	27
B.2	Càlcul del sistema d'elasticitat lineal	28
B.3	Càlcul del sistema de l'equació de dany	30
B.4	Càlcul de l'energia dissipada	31
B.5	Càlcul de l'energia de deformació	32
B.6	Càlcul dels valors nodals de la background mesh	33

1. Introducció

El camp de la mecànica de fractures ha estat d'interès al llarg del temps en l'àmbit de l'enginyeria. Entendre i predir els processos de fractura en materials i estructures té rellevància en molts dels problemes d'enginyeria.

Essencialment, hi ha dos tipus d'estratègies per modelar la fractura en materials fràgils: descripcions discontinues o contínues del crack. Els models discontinus descriuen el crack a través d'un camp de desplaçaments discontinu. Tenen la limitació, però, de no tenir una estratègia rigorosa per determinar la direcció de propagació del crack a cada instant de temps. Els models continus, en canvi, compten amb un camp de desplaçaments continu i descriuen el crack a través d'una zona danyada. Dintre del grup de models continus hi ha els models phase-field i els gradient-damage. En aquest treball, ens centrarem en models de phase-field, que venen donats per una formulació energètica del problema.

En aquesta línia, al 1920, Griffith va proposar una aproximació energètica del problema. Va establir que el procés de fractura avançarà per tal de minimitzar l'energia total del cos, que es pot escriure com la suma de l'energia elàstica de deformació i l'energia de generació de la fractura. En aquest sentit, podem entendre el procés com una competició entre les dues energies. És a dir, el crack avançarà si l'energia despesa per la propagació del crack compensa l'energia gastada en crear-lo. Des de llavors, la teoria de Griffith ha servit de base per a desenvolupar més models de fractura.

En particular, en aquest treball considerarem un model de fractura de phase-field dinàmic. Els models de phase-field per a fractura fràgil han sigut objecte d'estudi tant des del punt de vista teòric com computacional. Aquests tracten les fractures o cracks del material com a zones danyades que han perdut rigidesa. Per representar aquestes zones, el model introdueix un camp escalar -anomenat dany o variable phase-field- que varia de manera suau entre dos valors associats als estats intacte i trencat. L'evolució d'aquest camp com a resultat de les forces externes modela el procés de fractura. Matemàticament, el model consisteix en un sistema d'equacions no lineals acoblades, una que descriu l'equilibri de tensions i una altra que modela l'evolució de la variable phase-field. El que fa especialment atractiva aquesta aproximació i pel que s'ha popularitzat és que l'evolució de les fractures està incorporada a les equacions i processos complexos com la iniciació del crack, la seva propagació, bifurcacions, etc. s'obtenen de manera natural resolent el sistema d'equacions.

Els models phase-field es formulen a partir de la minimització de l'energia total del cos. Francfort i Marigo [2] van donar una expressió per a l'energia total, que va ser adaptada a [1] per a poder ser tractada numèricament. Els models phase-field que trobem a la literatura poden ser quasi-estàtics o dinàmics. Aquests últims tenen l'avantatge que permeten fer una interpretació més física dels resultats pel fet de considerar la variable temporal.

1.1 Estructura i objectius del treball

L'objectiu principal del treball és desenvolupar una eina computacional basada en un model de phase-field dinàmic per a fractures de materials fràgils i contrastar-lo mitjançant un parell de problemes clàssics dins de la teoria de la fractura: un problema de desplaçaments imposats i un altre de ramificació de fractures.

Dintre d'aquest últim es fa una anàlisi comparativa per estudiar la influència del paràmetre material G_c en la propagació del crack. També es marca com a objectiu la capacitat de realitzar un experiment numèric de manera completa. Això inclou aspectes de preprocés (com la creació i actualització de la malla), de programació i de postprocés (com l'elaboració adequada de gràfics).

Respecte a l'estructura del treball, es comença presentant el model, definint tots els conceptes necessaris fins a establir el sistema d'equacions que el constitueix. A continuació, es presenten i es mostren els resultats dels dos problemes esmentats anteriorment. En el següent apartat, s'explica l'esquema numèric emprat per resoldre el sistema d'equacions del model, fent èmfasi en cada una de les equacions que el constitueix. Per últim, es mostra quin procés s'ha seguit per elaborar les malles, necessàries per discretitzar els dominis dels problemes.

Tots els codis utilitzats en aquest treball s'han programat en Matlab i es troben a l'apèndix [B](#). A causa de l'elevat cost computacional (tant en temps com en espai) que suposen aquests problemes, s'han hagut d'enviar els càlculs al clúster del LaCàn de la UPC.

2. Model de Phase-Field dinàmic

Considerem un cos $\Omega \subset \mathbb{R}^2$ amb una discontinuïtat interna Γ_c , que representa una fractura (també anomenada crack) del material (Figura 1a). Sota deformació, el procés de fractura es dona per tal de minimitzar l'energia total del cos, que es pot expressar com la suma de l'energia elàstica de deformació i de l'energia de generació del crack. És a dir,

$$E(\mathbf{u}, d) = \int_{\Omega} \Psi(\epsilon) d\Omega + G_c \int_{\Gamma_c} d\Gamma_c, \quad (1)$$

sent Ψ la densitat de l'energia elàstica i G_c la taxa d'alliberació d'energia crítica, un valor que en assolir-se, permet a la fractura avançar.

En el nostre cas, ens restringirem a materials elàstics lineals isòtrops homogenis. Les tensions σ_{ij} estan doncs relacionades linealment amb les deformacions ϵ_{ij} , $\sigma_{ij} = \sum_{k,l} C_{ijkl} \epsilon_{kl}$, amb \mathcal{C} un tensor d'elasticitat de quart ordre que depèn dels paràmetres de Lamé del material. De fet, només en depèn de dos, ja que per ser el material d'aquestes característiques, la resta de paràmetres es poden escriure en funció d'aquests. Així, sota aquestes hipòtesis, la densitat d'energia elàstica ve donada per $\Psi(\epsilon) = \epsilon : \mathcal{C} : \epsilon$, on ϵ és el tensor de petites deformacions, definit a partir dels desplaçaments \mathbf{u} com $\epsilon(\mathbf{u}) = (\nabla \mathbf{u} + \nabla \mathbf{u}^T)/2$. El símbol $:$ denota l'operador de doble contracció que, per exemple, s'aplica com $(\mathcal{C} : \epsilon)_{ij} = \sum_{k,l} C_{ijkl} \epsilon_{kl}$.

Per tal de facilitar el tractament numèric de (1), en el model de phase-field s'introdueix una representació contínua del crack Γ_c (vegeu [1]). El crack ve definit ara per un camp $d(\mathbf{x}, t)$, que varia suaument entre els dos valors que representen els estats intacte i trencat, 0 i 1 respectivament (Figura 1b). Se l'anomena variable phase-field o variable de dany. Així doncs, l'energia total (1) s'aproxima per

$$E_l(\mathbf{u}, d) = \int_{\Omega} (1 - d)^2 \Psi(\epsilon) d\Omega + G_c \int_{\Omega} \left(\frac{d^2}{2l} + \frac{l}{2} |\nabla d|^2 \right) d\Omega, \quad (2)$$

on l regula l'amplada del crack.

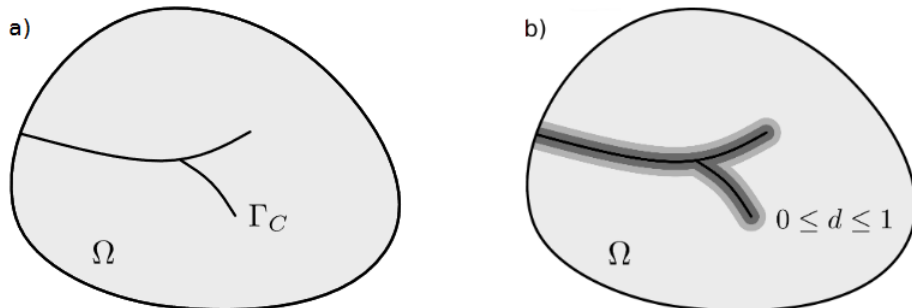


Figura 1: a) Cos amb un crack Γ_c . b) Cos amb la representació del crack mitjançant d . Figura extreta de [6].

Establint balanç de moments (vegeu [7]) i minimitzant el funcional d'energia (2), s'obté el sistema a

solucionar pel desplaçament \mathbf{u} , i la variable de phase-field d ,

$$\begin{cases} \rho \ddot{\mathbf{u}} = \nabla \cdot \boldsymbol{\sigma}, \\ -l^2 \Delta d + d = \frac{2l}{G_c} (1-d) \mathcal{H}, \end{cases} \quad (3)$$

sent $\ddot{\mathbf{u}} = \frac{d^2 \mathbf{u}}{dt^2}$ l'acceleració i $\boldsymbol{\sigma}$ el tensor de tensions de Cauchy definit per

$$\boldsymbol{\sigma}(\mathbf{u}, d) = (1-d)^2 \frac{\partial \Psi(\varepsilon)}{\partial \varepsilon} = (1-d)^2 \mathcal{C} : \varepsilon(\mathbf{u}).$$

El sistema (3) es resol amb les condicions de contorn

$$\begin{aligned} \mathbf{u} &= \mathbf{u}_D & \text{a } \Gamma_D, \\ \boldsymbol{\sigma} \cdot \mathbf{n} &= \mathbf{t}_N & \text{a } \Gamma_N, \\ \nabla d \cdot \mathbf{n} &= 0 & \text{a } \partial\Omega, \end{aligned}$$

sent \mathbf{u}_D i \mathbf{t}_N els desplaçaments i traccions imposades, respectivament. Addicionalment, considerem les condicions inicials

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}^0(\mathbf{x}), \quad \dot{\mathbf{u}}(\mathbf{x}, 0) = \dot{\mathbf{u}}^0(\mathbf{x}), \quad d(\mathbf{x}, 0) = d^0(\mathbf{x}) \text{ a } \Omega.$$

Notem que a la segona equació de (3) hem substituït Ψ per una variable \mathcal{H} definida per $\mathcal{H}(\mathbf{x}, t) := \max_{\tau \in (0, t)} \Psi(\mathbf{x}, \tau)$, fent que el crack sigui irreversible (vegeu [5]).

3. Test de l'L

Considerem una peça en forma d'L (Figura 2), discretitzada mitjançant elements finits, que està subjecta per la part inferior. És a dir, imposem que el desplaçament sigui 0 ($\mathbf{u} = 0$) als nodes de la frontera inferior (encerclats a la figura). L'altre node encerclat de la figura és el (220,0), on imposem un desplaçament vertical $u_y = u_D(t)$ depenent del temps, que és el que fa que la peça es deformi i es trenqui. Aquest desplaçament $u_D(t)$ depèn quadràticament del temps els primers segons, és a dir, estem aplicant una acceleració, per després mantenir-se amb una velocitat constant fins al final de l'experiment ($u_D(t)$ lineal en t).

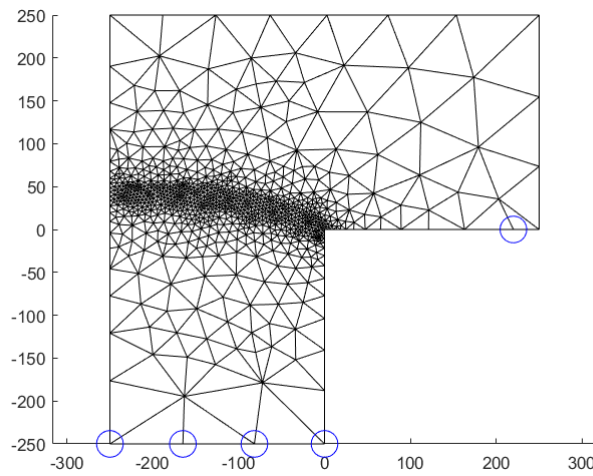


Figura 2: Discretització d'una peça en forma de L usant una malla amb elements triangulars.

Per resoldre el problema utilitzem una malla adaptada que com es pot veure a la Figura 2, és més fina allà on sabem que avançarà el crack (vegeu [6]). Aquesta no és la situació en general, on cal anar refinant la malla a mesura que avança el crack.

Els paràmetres materials emprats en l'experiment són els següents: $E = 25.8423$ GPa, $\nu = 0.18$, $\rho = 2350$ kg/m³, $G_c = 8.9 \cdot 10^{-5}$ kN/mm² i $l = 10$ mm. Com a condicions inicials, considerem $\mathbf{u}^0 = 0$, $\dot{\mathbf{u}}^0 = 0$ i $d^0 = 0$. Per a l'equació de dany, les condicions de contorn són totes Neumann homogènies. En canvi, per a l'equació d'elasticitat lineal tenim condicions de Dirichlet relacionades amb les dues condicions descrites anteriorment: la subjecció inferior i el desplaçament imposat. Més concretament, tenim que

$$u_D(t) = \begin{cases} \frac{1}{2}(1 \cdot 10^{-2})t^2 & t \leq 0.1 \\ (1 \cdot 10^{-3})t - 5 \cdot 10^{-5} & t \geq 0.1 \end{cases}$$

El sistema s'ha integrat fins a un temps final de 400 s i s'han utilitzat 4 passos de temps diferents: $\Delta t = 0.4$, $\Delta t = 4 \cdot 10^{-2}$ i $\Delta t = 4 \cdot 10^{-3}$. La Figura 3, mostra la solució per $\Delta t = 4 \cdot 10^{-3}$. S'observa com es propaga el crack alhora que es veu com es va deformant la peça al llarg del temps.

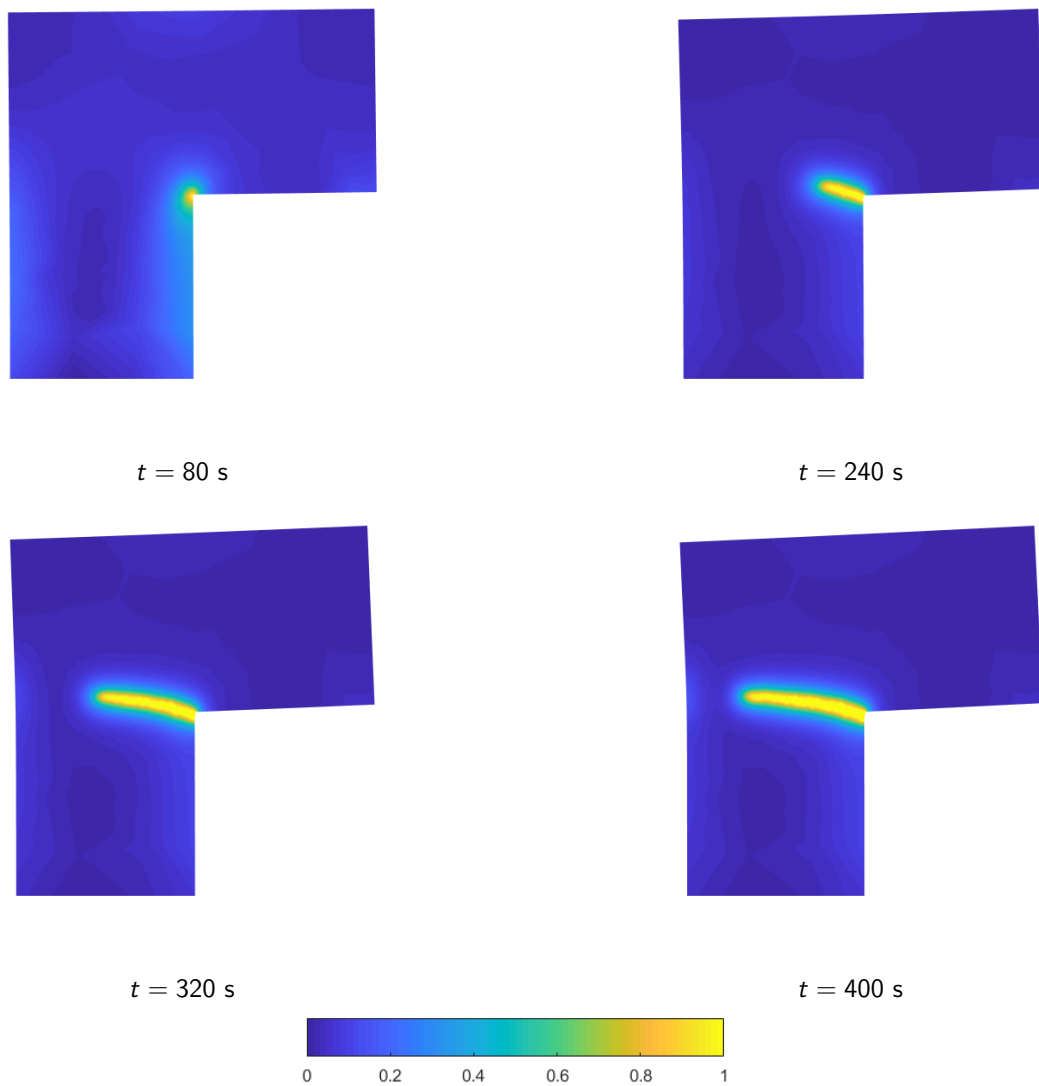


Figura 3: Evolució del dany al llarg del temps. Els desplaçaments han estat escalats per un factor de 50.

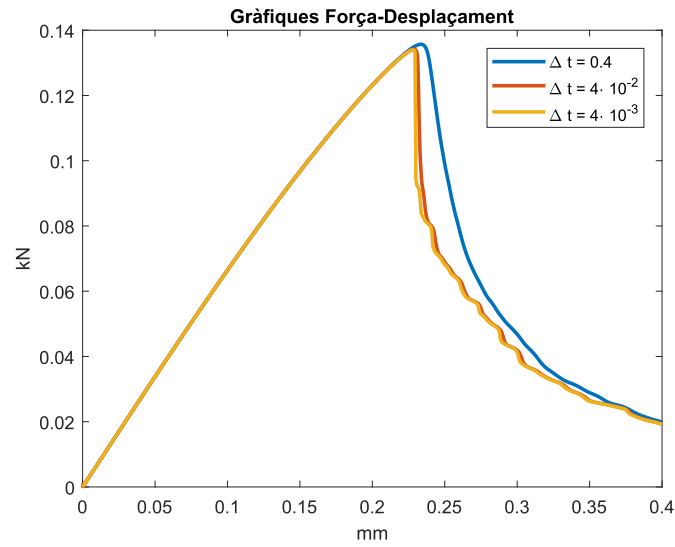


Figura 4: Gràfiques Força-Desplaçament per a diferents Δt .

A més, per a cada pas de temps, s'ha dibuixat la gràfica força-desplaçament (Figura 4). Aquesta representa la força que s'hauria d'aplicar a la peça per tal d'assolir el desplaçament imposat u_D . Es pot veure que al principi les corbes segueixen un comportament lineal, és a dir, la força necessària és proporcional al desplaçament desitjat. A partir d'un cert punt, però, les corbes comencen a decreixer ràpidament. Això es deu al fet que la peça s'ha començat a trencar, i per tant, la força necessària per aconseguir un determinat desplaçament disminueix. A part de la interpretació física, s'observa convergència en temps, ja que a mesura que disminuïm el pas de temps, les corbes són més semblants entre elles.

4. Crack branching dinàmic

El crack branching (bifurcació del crack) és un fenomen freqüentment observat en sòlids fràgils. Es produeix quan el material està sotmès a grans tensions que no li permeten dissipar tota l'energia en un sol crack.

L'objectiu d'aquesta secció és usar el model de phase-field dinàmic per simular un test de crack branching. Usarem com a referència, tant per les mides de la peça com pels paràmetres, l'exemple de branching dinàmic mostrat a [3]. Addicionalment, es fa una anàlisi de com afecta la variació del paràmetre G_c a la bifurcació.

4.1 Test de crack branching dinàmic

Considerem una peça rectangular de dimensions 100×40 mm, a la qual se li aplica una tensió de $\sigma = 1$ MPa als costats inferior i superior, i que es manté constant durant tota la simulació (Figura 5). El domini conté un crack inicial horitzontal que s'expandeix des del centre del costat esquerre fins al centre de la peça. Es modela amb una condició inicial $d^0 = 1$ a una banda d'amplada 1.25 mm.

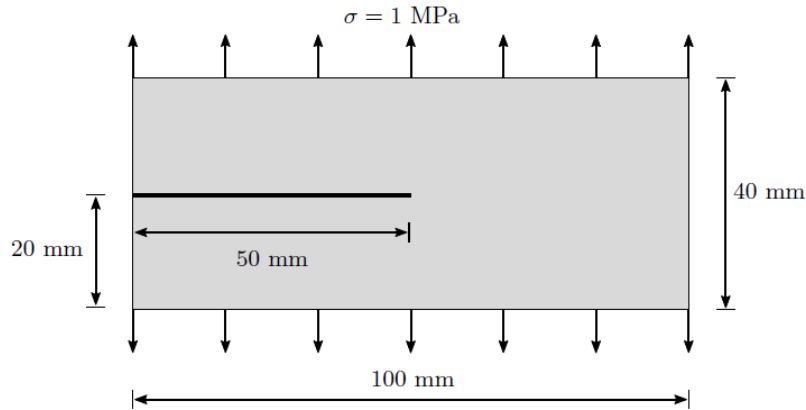


Figura 5: Geometria de la peça juntament amb les condicions de contorn del problema. Figura extreta de [3].

Per resoldre el problema, utilitzem una malla d'elements finits triangulars adaptada al crack inicial (Figura 6), sent $h_{min} = 0.5$ mm la mida d'element més petita. A més, és simètrica respecte l'eix $y = 0$ per tal de preservar la simetria de la solució numèrica del problema. A la secció 6, s'explica en detall el procediment de construcció de la malla, que es va adaptant a l'avançament del crack.

Els paràmetres materials emprats són $E = 32$ GPa, $\nu = 0.2$, $G_c = 6 \cdot 10^{-6}$ kN/mm², $\rho = 2450$ kg/m³ i $l = 1.25$ mm. Com a condicions inicials, considerem $\mathbf{u}^0 = 0$, $\dot{\mathbf{u}}^0 = 0$ i $d^0 = 0$ arreu excepte al crack inicial, on $d^0 = 1$. El crack inicial es modela com una condició de Dirichlet per a l'equació de dany. Per tot instant de temps, imposem $d = 1$ als nodes amb $x \leq 0$ i $|y| \leq 0.625$. A la Figura 7 es veu com amb una sola iteració, el crack inicial pren la forma característica del model de Phase-Field. D'altra banda, les tensions a les quals està sotmesa la peça s'imposen mitjançant condicions de Neumann per a l'equació d'elasticitat lineal. En concret, tenim que $\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{t}_N$ als costats inferior i superior, amb $\mathbf{t}_N = \begin{bmatrix} 0 \\ \pm 1 \end{bmatrix}$.

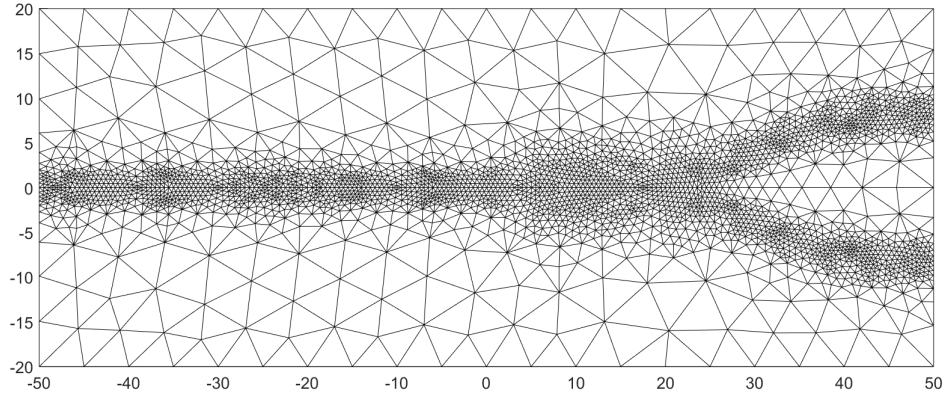


Figura 6: Discretització de la peça usant una malla amb elements triangulars. La malla s'ajusta durant el càlcul per adaptar-se a l'avançament del crack.

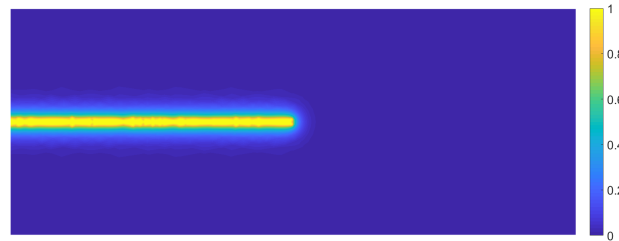


Figura 7: Variable de phase-field després d'una iteració (representació del crack inicial).

El sistema s'integra fins a $t = 0.004$ s, amb 5 passos de temps diferents: $\Delta t = 8 \cdot 10^{-6}$, $\Delta t = 4 \cdot 10^{-6}$, $\Delta t = 2 \cdot 10^{-6}$, $\Delta t = 1 \cdot 10^{-6}$ i $\Delta t = 5 \cdot 10^{-7}$. La Figura 8 mostra la solució per $\Delta t = 5 \cdot 10^{-7}$. S'observa com avança el crack i com es va deformant la peça al llarg del temps. El model captura perfectament la bifurcació que es produeix.

Per finalitzar, fem un estudi de les energies implicades en el problema, una anàlisi bastant habitual en aquests casos. D'una banda, tenim l'energia dissipada, E_{dis} , que és l'energia que emet el material pel fet de trencar-se. D'altra banda, l'energia elàstica de deformació, E_{elas} , és aquella emmagatzemada pel material pel fet d'estar sota un procés de deformació. Ambdues es poden calcular fent una integral sobre el domini de la següent manera:

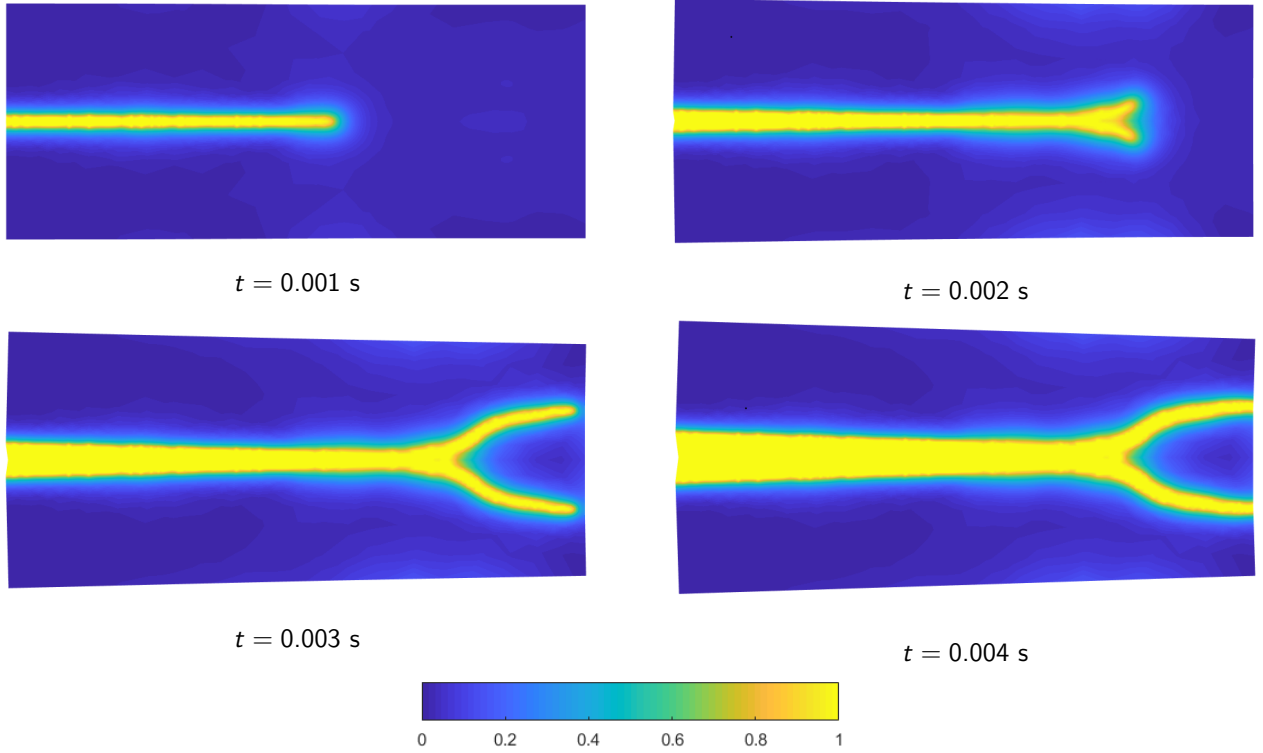


Figura 8: Evolució del dany al llarg del temps. Els desplaçaments han estat escalats per un factor de 20.

$$E_{dis} = \int_{\Omega} \left(\frac{d^2}{2l} + \frac{l}{2} |\nabla d|^2 \right) d\Omega,$$

$$E_{elas} = \int_{\Omega} (1 - d)^2 \Psi(\varepsilon) d\Omega.$$

La Figura 9a mostra l'energia dissipada al llarg del temps, que creix a mesura que va avançant el crack i es fa constant just al final quan la peça està acabada de trencar. En canvi, l'energia elàstica de deformació (Figura 9b) té diversos pics. Al principi, quan la peça no s'ha començat a trencar, E_{elas} creix perquè es comença a acumular energia a causa de les deformacions provocades per les tensions a les quals està sotmès el material. Quan la peça es comença a trencar, E_{elas} decreix perquè aquesta energia acumulada es dissipa, i així successivament. A més, les dues figures en serveixen per observar que hi ha convergència en temps, degut a semblança de les corbes en reduir el pas de temps.

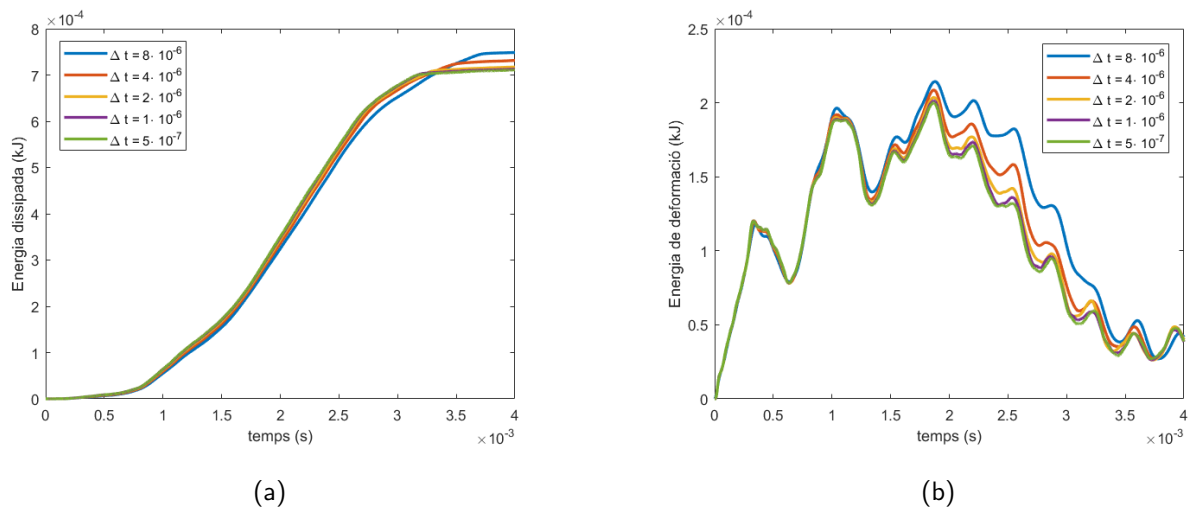


Figura 9: Evolució de (a) l'energia dissipada pel crack i (b) l'energia elàstica de deformació al llarg del temps per diferents Δt .

4.2 Anàlisi de G_c

Es vol estudiar ara quin impacte té el paràmetre G_c en la solució. Recordem que és el valor d'energia crítica que permet avançar al crack. Per fer-ho, resollem el problema de l'apartat anterior amb els mateixos paràmetres, però fent variar G_c . En particular, considerem els següents valors de la constant: $G_c = 5 \cdot 10^{-7}$ kN/mm², $G_c = 3 \cdot 10^{-6}$ kN/mm², $G_c = 6 \cdot 10^{-6}$ kN/mm² i $G_c = 1 \cdot 10^{-4}$ kN/mm². A la Figura 10 veiem l'aspecte final del cracks. S'observa com per valors grans ($G_c = 1 \cdot 10^{-4}$ kN/mm²) la bifurcació es produeix abans en espai i les branques formen un angle major. A mesura que fem G_c petit, el punt on bifurca el crack està més allunyat del centre de la peça i les branques tendeixen a tancar-se cada cop més, fins al punt d'arribar a no bifurcar ($G_c = 5 \cdot 10^{-7}$ kN/mm²). Aquest fenomen s'explica de la següent manera: com la G_c és més gran, el material acumula més energia de la que pot dissipar en un sol crack i necessita bifurcar abans.

En el gràfic de l'energia dissipada (Figura 11), s'observa que els cracks associats a G_c més grans triguen més temps a avançar. Això és conseqüència de la pròpia definició de la constant, com més gran sigui, més temps es necessita per acumular l'energia necessària perquè el crack avanci.

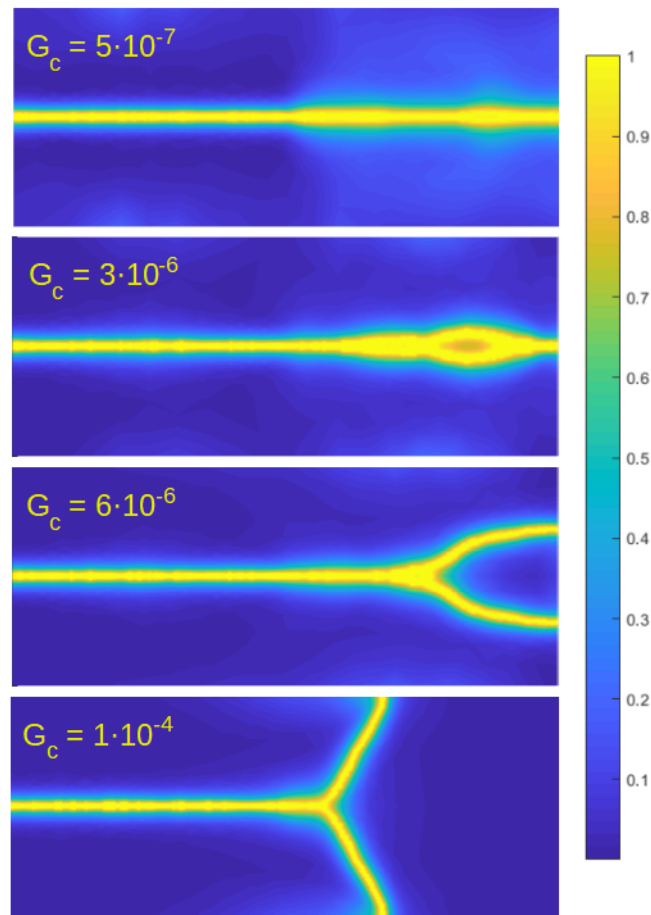


Figura 10: Dany de la peça per diferents valors de G_c .

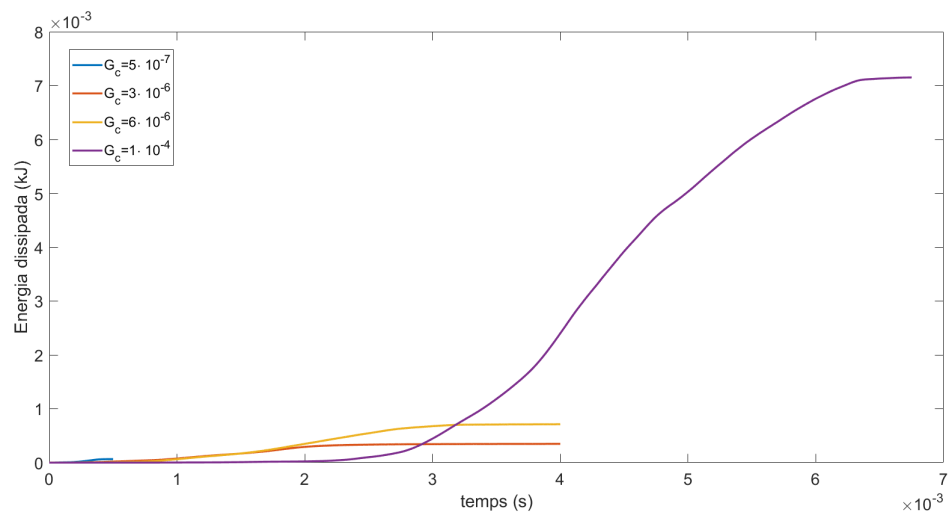


Figura 11: Evolució de l'energia dissipada al llarg del temps per diferents valors de G_c .

5. Mètode numèric

En aquesta secció, es mostra l'esquema numèric emprat per tal de resoldre el sistema (3) en cadascun dels exemples mostrats anteriorment. Posteriorment, com cada equació del sistema es resol independentment, s'analitza cada una en un apartat i se'n comprova la convergència per a problemes amb solució analítica coneguda.

5.1 Esquema numèric

El sistema (3) està format per dues equacions no lineals acoblades. Necessitem un algorisme que donada la solució (\mathbf{u}^n, d^n) a temps n , trobi la solució $(\mathbf{u}^{n+1}, d^{n+1})$ a temps $n + 1$. Una aproximació interessant és intentar trobar els dos camps simultàniament mitjançant el mètode de Newton. L'inconvenient és que el mètode de Newton clàssic no té per què convergir i s'haurien de fer adaptacions per tal que trobés la solució (vegeu [8]).

Una alternativa, potser no tan eficient, però si més fàcilment implementable, és solucionar les dues equacions per separat. Per la primera equació, aproximem d^{n+1} per la solució a l'instant anterior, d^n , i resollem per trobar \mathbf{u}^{n+1} . Posteriorment, resollem la segona equació i trobem d^{n+1} . L'avantatge d'aquesta estratègia és que els problemes desacoblats són lineals. L'inconvenient és que al fer l'aproximació $d^{n+1} \approx d^n$, l'ordre en temps és $\mathcal{O}(\Delta t)$, i cal utilitzar més passos de temps per trobar una solució suficientment precisa.

Més concretament, donada la solució (\mathbf{u}^n, d^n) a temps n seguim el següent algorisme per trobar la solució a temps $n + 1$:

1. Donat d^n , calcular \mathbf{u}^{n+1} resolent l'equació lineal

$$\rho \ddot{\mathbf{u}}^{n+1} = \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}^{n+1}, d^n)$$

amb condicions de contorn $\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{t}_N^{n+1}$ a Γ_N i $\mathbf{u} = \mathbf{u}_D^{n+1}$ a Γ_D .

2. Actualitzar \mathcal{H} utilitzant \mathbf{u}^{n+1} :

$$\mathcal{H}^{n+1} = \max(\mathcal{H}^n, \Psi(\mathbf{u}^{n+1})).$$

3. Donat \mathcal{H}^{n+1} , calcular d^{n+1} resolent l'equació

$$-l^2 \Delta d^{n+1} + d^{n+1} = \frac{2l}{G_c} (1 - d^{n+1}) \mathcal{H}^{n+1}$$

amb condició de contorn $\nabla d^{n+1} \cdot \mathbf{n} = 0$ a $\partial\Omega$.

A continuació, es presenten els mètodes numèrics emprats per a la solució dels problemes desacoblats, corresponents als punts 1 i 3 anteriors.

5.2 Equació d'Elasticitat Lineal Transitòria

Siguin \mathbf{u} el camp de desplaçaments i $\boldsymbol{\sigma}$ el tensor de tensions de Cauchy, l'equació d'elasticitat lineal transitòria té la següent forma:

$$\rho \ddot{\mathbf{u}} = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b} \quad (4)$$

$$\boldsymbol{\sigma}(\mathbf{u}, d) = (1 - d)^2 \mathcal{C} : \boldsymbol{\varepsilon}(\mathbf{u}),$$

on \mathbf{b} són les forces externes del sistema, ρ és la densitat del material, d és la variable de Phase-Field, \mathcal{C} un tensor de 4t ordre depenent dels paràmetres materials i $\boldsymbol{\varepsilon}(\mathbf{u})$ el tensor de petites deformacions.

En el cas 2-dimensional, amb $\mathbf{u} = \begin{bmatrix} u_x \\ u_y \end{bmatrix}$, el tensor de petites deformacions ve donat per la següent expressió:

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \nabla^s \mathbf{u} = \begin{bmatrix} \frac{\partial u_x}{\partial x} & \frac{1}{2} \left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \right) \\ \frac{1}{2} \left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \right) & \frac{\partial u_y}{\partial y} \end{bmatrix}.$$

Donat que el tensor és simètric, podem usar la notació de Mandel i treballar amb el vector

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \begin{bmatrix} \frac{\partial u_x}{\partial x} \\ \frac{\partial u_y}{\partial y} \\ \frac{\sqrt{2}}{2} \left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \right) \end{bmatrix}.$$

Aquesta notació té l'avantatge que permet utilitzar les operacions convencionals usades amb vectors. Per exemple, sota la hipòtesi de deformació plana, l'equació constitutiva es pot escriure com

$$\boldsymbol{\sigma}(\mathbf{u}, d) = (1 - d)^2 \mathcal{C} \boldsymbol{\varepsilon}(\mathbf{u})$$

amb

$$\mathcal{C} = \frac{E}{(1 + \nu)(1 - 2\nu)} \begin{bmatrix} 1 - \nu & \nu & 0 \\ \nu & 1 - \nu & 0 \\ 0 & 0 & 1 - 2\nu \end{bmatrix},$$

on ν és el coeficient de Poisson i E el mòdul de Young.

Plantegem a continuació la solució numèrica de (4), suposant d conegut i resolent per \mathbf{u} , amb les condicions de contorn següents:

$$\begin{aligned} \mathbf{u} &= \mathbf{u}_D & \text{a } \Gamma_D \\ \boldsymbol{\sigma} \cdot \mathbf{n} &= \mathbf{t}_N & \text{a } \Gamma_N \end{aligned} \quad (5)$$

on $\Gamma_D \cup \Gamma_N = \partial\Omega$, i u_D i t_N són desplaçaments i traccions donades.

La forma feble del problema és la següent (la derivació es pot trobar a l'apèndix A): Trobar $\mathbf{u} \in [H^1(\Omega)]^2$ amb $\mathbf{u} = \mathbf{u}_D$ a Γ_D i tal que

$$\int_{\Omega} \rho \mathbf{v} \cdot \ddot{\mathbf{u}} \, d\Omega + \int_{\Omega} (1 - d)^2 \boldsymbol{\varepsilon}(\mathbf{v}) \cdot \mathcal{C} : \boldsymbol{\varepsilon}(\mathbf{u}) \, d\Omega = \int_{\Omega} \rho \mathbf{v} \cdot \mathbf{b} \, d\Omega + \int_{\Gamma_N} \mathbf{v} \cdot \mathbf{t}_N \, d\Gamma_N, \quad \forall \mathbf{v} \in [H^1(\Omega)]^2 \quad (6)$$

amb $\mathbf{v} = 0$ a Γ_D

Per resoldre el problema, discretitzem el domini Ω i fem una aproximació amb elements finits de \mathbf{u} usant funcions de forma $\{N_i\}_{i=1}^n$. Tenim doncs

$$\mathbf{u}(x) \approx \sum_{i=1}^n \begin{bmatrix} (u_x)_i \\ (u_y)_i \end{bmatrix} N_i(x),$$

on n és el nombre de nodes de la malla.

Per simplicitat i qüestions d'implementació, en comptes d'utilitzar un sumatori és convenient expressar \mathbf{u} usant matrius de la següent manera:

$$\mathbf{u} = \left[\begin{array}{ccc|ccc} N_1 & \dots & N_n & 0 & 0 & 0 \\ 0 & & & N_1 & \dots & N_n \end{array} \right] \begin{bmatrix} (u_x)_1 \\ (u_y)_1 \\ \vdots \\ (u_x)_n \\ (u_y)_n \end{bmatrix} =: [\mathbf{vecN}] \mathbf{U}.$$

Anàlogament, $\ddot{\mathbf{u}} = [\mathbf{vecN}] \ddot{\mathbf{U}}$, on \mathbf{U} i $\ddot{\mathbf{U}}$ són els vectors de valors nodals del desplaçament i de les acceleracions respectivament. Amb aquesta notació, el tensor de petites deformacions, $\varepsilon(\mathbf{u})$, s'escriu com

$$\begin{aligned} \varepsilon(\mathbf{u}) &= \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\sqrt{2}}{2} \frac{\partial}{\partial y} & \frac{\sqrt{2}}{2} \frac{\partial}{\partial x} \end{bmatrix} \mathbf{u} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\sqrt{2}}{2} \frac{\partial}{\partial y} & \frac{\sqrt{2}}{2} \frac{\partial}{\partial x} \end{bmatrix} [\mathbf{vecN}] \mathbf{U} = \\ &= \left[\begin{array}{ccc|ccc} \frac{\partial N_1}{\partial x} & \dots & \frac{\partial N_n}{\partial x} & 0 & 0 & 0 \\ 0 & & & \frac{\partial N_1}{\partial y} & \dots & \frac{\partial N_n}{\partial y} \\ \frac{\sqrt{2}}{2} \frac{\partial N_1}{\partial y} & \dots & \frac{\sqrt{2}}{2} \frac{\partial N_n}{\partial y} & \frac{\sqrt{2}}{2} \frac{\partial N_1}{\partial x} & \dots & \frac{\sqrt{2}}{2} \frac{\partial N_n}{\partial x} \end{array} \right] \mathbf{U} =: [\mathbf{strainN}] \mathbf{U}. \end{aligned}$$

Prenent $\mathbf{v} = [\mathbf{vecN}] \mathbf{V}$ i substituint a la forma feble (6), s'obté

$$\begin{aligned} \mathbf{V}^T \left[\int_{\Omega} \rho [\mathbf{vecN}]^T [\mathbf{vecN}] d\Omega \right] \ddot{\mathbf{U}} + \mathbf{V}^T \left[\int_{\Omega} (1-d)^2 [\mathbf{strainN}]^T \mathcal{C} [\mathbf{strainN}] d\Omega \right] \mathbf{U} = \\ = \mathbf{V}^T \left[\int_{\Omega} \rho [\mathbf{vecN}]^T \mathbf{b} d\Omega + \int_{\Gamma_N} [\mathbf{vecN}]^T \mathbf{t}_N d\Gamma_N \right], \quad \forall \mathbf{V} \in \mathbb{R}^{2n}. \end{aligned}$$

D'aquesta expressió resulta el sistema

$$\mathbf{M} \ddot{\mathbf{U}} + \mathbf{K} \mathbf{U} = \mathbf{f} \quad (7)$$

amb

$$\mathbf{M} = \int_{\Omega} \rho [\mathbf{vecN}]^T [\mathbf{vecN}] d\Omega, \quad \mathbf{K} = \int_{\Omega} (1-d)^2 [\mathbf{strainN}]^T \mathcal{C} [\mathbf{strainN}] d\Omega$$

$$\mathbf{f} = \int_{\Omega} \rho [\mathbf{vecN}]^T \mathbf{b} \, d\Omega + \int_{\Gamma_N} [\mathbf{vecN}]^T \mathbf{t}_N \, d\Gamma_N,$$

on encara no s'han imposat les condicions de Dirichlet.

5.3 Mètode de Newmark

El sistema d'EDOS de segon d'ordre (7) es pot resoldre utilitzant el mètode de Newmark, que és molt emprat en dinàmica d'estructures.

Sigui Δt un pas de temps fixat. Donada la solució a $t^n = n\Delta t$, \mathbf{U}^n , $\dot{\mathbf{U}}^n$ i $\ddot{\mathbf{U}}^{n+1}$, la idea del mètode és utilitzar sèries de Taylor generalitzades per a \mathbf{U}^{n+1} i $\dot{\mathbf{U}}^{n+1}$,

$$\begin{aligned} \mathbf{U}^{n+1} &= \mathbf{U}^n + \Delta t \dot{\mathbf{U}}^n + \Delta t^2 \left(\left(\frac{1}{2} - \beta \right) \ddot{\mathbf{U}}^n + \beta \ddot{\mathbf{U}}^{n+1} \right) \\ \dot{\mathbf{U}}^{n+1} &= \dot{\mathbf{U}}^n + \Delta t \left((1 - \gamma) \ddot{\mathbf{U}}^n + \gamma \ddot{\mathbf{U}}^{n+1} \right) \end{aligned} \quad (8)$$

que depenen de dos paràmetres, β i γ , tals que $0 \leq \beta \leq 1/2$ i $0 \leq \gamma \leq 1$. Notem que per $\beta = \gamma = 0$ tenim les sèries de Taylor habituals.

Ara, establim equilibri a temps t^{n+1} per a l'equació:

$$\mathbf{M} \ddot{\mathbf{U}}^{n+1} + \mathbf{K} \mathbf{U}^{n+1} = \mathbf{f}^{n+1}$$

Substituïm \mathbf{U}^{n+1} i $\dot{\mathbf{U}}^{n+1}$ usant (8), reordenem els termes i obtenim el sistema d'equacions:

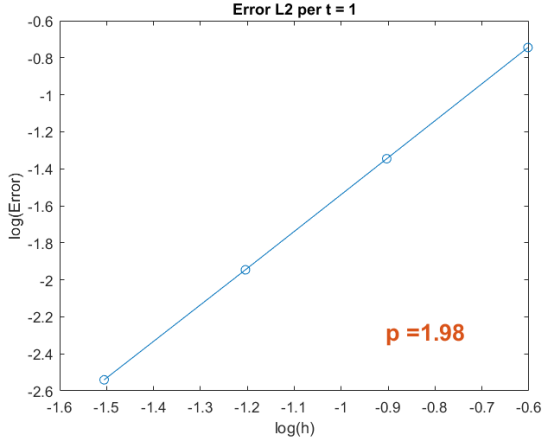
$$(\mathbf{M} + \Delta t^2 \beta \mathbf{K}) \ddot{\mathbf{U}}^{n+1} = \mathbf{f}^{n+1} - \mathbf{K} \left(\mathbf{U}^n + \Delta t \dot{\mathbf{U}}^n + \left(\frac{1}{2} - \beta \right) \Delta t^2 \ddot{\mathbf{U}}^n \right)$$

Per a cada pas de temps s'ha de resoldre un sistema d'equacions lineal per calcular les acceleracions $\ddot{\mathbf{U}}^{n+1}$. Posteriorment, es calculen les velocitats $\dot{\mathbf{U}}^{n+1}$, i els desplaçaments, \mathbf{U}^{n+1} , fent servir (8). Respecte a l'elecció de β i γ , usem $\beta = 1/4$, $\gamma = 1/2$, ja que llavors tenim estabilitat incondicional per a qualsevol Δt i la convergència és d'ordre $\mathcal{O}(\Delta t^2)$.

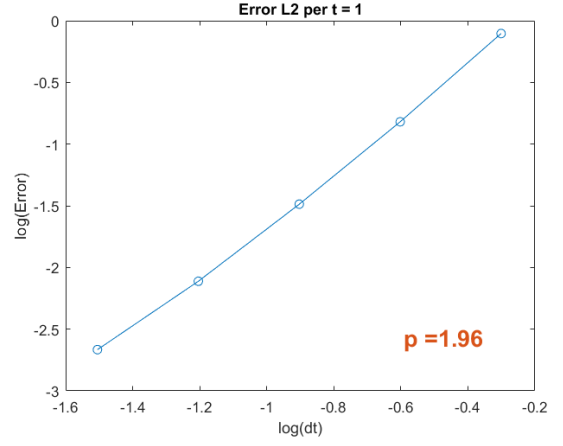
El mètode descrit s'ha implementat utilitzant Matlab i per garantir el seu bon funcionament, hem fet una anàlisi de convergència. Per fer-la, imposem que la solució sigui $u = t^4 \left[\frac{\sin(x+y)}{x^2 + y^2} \right]$. Amb aquesta, calculem analíticament el terme font \mathbf{f} per tal que l'equació (6) es compleixi, tenint en compte que els paràmetres escollits són $E = 1$, $\nu = 0.2$, $\rho = 1$ i $d(x, y) = (\sin(x+y) + 1)/5$ per tot instant de temps. Un cop fet això, podem trobar u numèricament.

Per analitzar la convergència, comparem la solució numèrica amb l'analítica calculant l'error en norma L^2 . Primer, estudiem la convergència en espai usant una aproximació amb elements finits lineals. Fixem un pas de temps Δt i anem fent cada cop la malla més fina. Com es veu en la Figura 12a, el pendent de la recta és $p = 1.98 \approx 2$, i per tant, la convergència és òptima, quadràtica en espai. Respecte al

temps, l'ordre de convergència també és quadràtic (Figura 12b), i per tant coincideix amb la convergència esperada pel mètode de Newmark.



(a) Convergència en espai ($\Delta t = 0.01$)



(b) Convergència en temps ($h = 1/64$)

Figura 12: Gràfiques de convergència amb el pendent de la recta p .

5.4 Equació de dany

L'equació que modelitza el dany d és

$$-l^2 \Delta d + d = \frac{2l}{G_c} (1 - d) \mathcal{H}, \quad (9)$$

on $\mathcal{H}(x, t) := \max_{\tau \in (0, t)} \Psi(x, \tau)$, sent Ψ la funció d'energia potencial deguda a la deformació. Aquesta depèn dels desplaçaments a temps t i ve donada per $\Psi(x, t) = \frac{1}{2} \varepsilon(\mathbf{u}(x, t)) \cdot \mathbf{C} \cdot \varepsilon(\mathbf{u}(x, t))$. Les condicions de contorn són Neumann homogènies a tota la vora.

La forma feble del problema és doncs: Trobar $d \in H^1(\Omega)$ tal que

$$\int_{\Omega} v \left(\frac{G_c}{l} + 2\mathcal{H} \right) d \, d\Omega + \int_{\Omega} G_c l \nabla v \cdot \nabla d \, d\Omega = \int_{\Omega} 2\mathcal{H} v \, d\Omega, \quad \forall v \in H^1(\Omega). \quad (10)$$

Per resoldre el problema, discretitzem el domini Ω i fem una aproximació de d amb elements finits, usant funcions de forma $\{N_i\}_{i=1}^n$, i.e, $d \approx \sum_{i=1}^n d_i N_i$. Prenem $v = N_i$, i substituïm a (10) obtenint

$$\sum_{j=1}^n \left[\int_{\Omega} \left(\frac{G_c}{l} + 2\mathcal{H} \right) N_i N_j + G_c l \nabla N_i \nabla N_j \, d\Omega \right] d_j = \int_{\Omega} 2\mathcal{H} N_i \, d\Omega, \quad \forall i = 1 \dots n.$$

D'aquesta expressió en resulta el sistema $\mathbf{Kd} = \mathbf{f}$ (sent \mathbf{d} el vector de valors nodals del dany) amb:

$$\mathbf{K}_{ij} = \int_{\Omega} \left(\frac{G_c}{l} + 2\mathcal{H} \right) N_i N_j + G_c l \nabla N_i \nabla N_j \, d\Omega, \quad \mathbf{f}_i = \int_{\Omega} 2\mathcal{H} N_i \, d\Omega.$$

Per solucionar l'equació del dany, s'implementa un codi de Matlab que calcula la matriu \mathbf{K} i el vector \mathbf{f} per després resoldre el sistema lineal $\mathbf{Kd} = \mathbf{f}$. Com abans, fem una prova de convergència per comprovar la correctesa del codi. Imposem que la solució sigui $d = \sin(\pi x)\sin(\pi y)$ i calculem analíticament \mathcal{H} per tal que l'equació (9) es compleixi. Els paràmetres utilitzats són $l = 1$ i $G_c = 2$.

A continuació, trobem la solució numèrica de d per a diverses malles i calculem l'error en norma L^2 respecte a l'analítica. La Figura 13 mostra que la convergència és d'ordre 2 en espai, tal com caldria esperar, ja que les funcions de forma considerades són lineals.

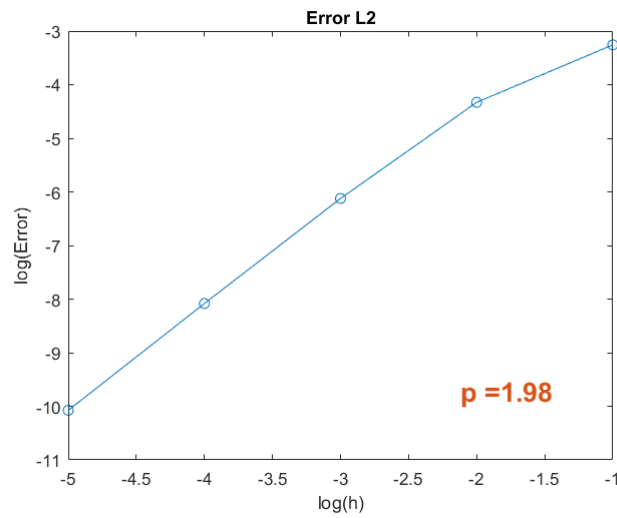


Figura 13: Gràfica de convergència amb el pendent de la recta p .

6. Creació de malles

En els problemes de fractura, un aspecte clau és el disseny de la malla per tal de fer una bona discretització del domini. És necessari que la malla sigui prou fina al voltant del crack per poder capturar correctament la seva evolució. Podem pensar que una solució és fer una discretització molt fina a tot el domini, però això comporta un alt cost computacional que no sempre es pot assolir. Per això, una estratègia molt utilitzada és refinar l'àrea al voltant del crack, i deixar una discretització més grollera a la resta. L'inconvenient, però, és que no se sap per on avançarà el crack i per tant no podem fer la malla completa directament. El que es fa és integrar el sistema durant un cert temps, veure per on avança el crack, refer la malla refinant aquella zona, tornar a integrar el sistema, i així successivament.

Idealment, voldríem un algorisme que fes aquest procés automàticament, per exemple, detectant allà on el dany d es fa proper a 1 i refinant aquella zona. Aquests algorismes però, no són trivials d'implementar. Per això, en aquest projecte, el procés s'ha fet de manera manual, amb el cost en temps que això comporta. En els següents apartats, s'explica tot aquest procediment, exemplificant-lo amb la malla creada pel problema de la secció 4.

6.1 EZ4U i Background Mesh

EZ4U és un software generador de malles creat pel LaCàn. Permet dissenyar malles de forma interactiva i exportar-les a un arxiu de text, per després utilitzar-les en entorns diversos (Matlab en el nostre cas). A la Figura 14 es mostra l'entorn de treball del EZ4U, amb les diferents barres d'eines i la finestra que permet visualitzar la malla dins del domini.

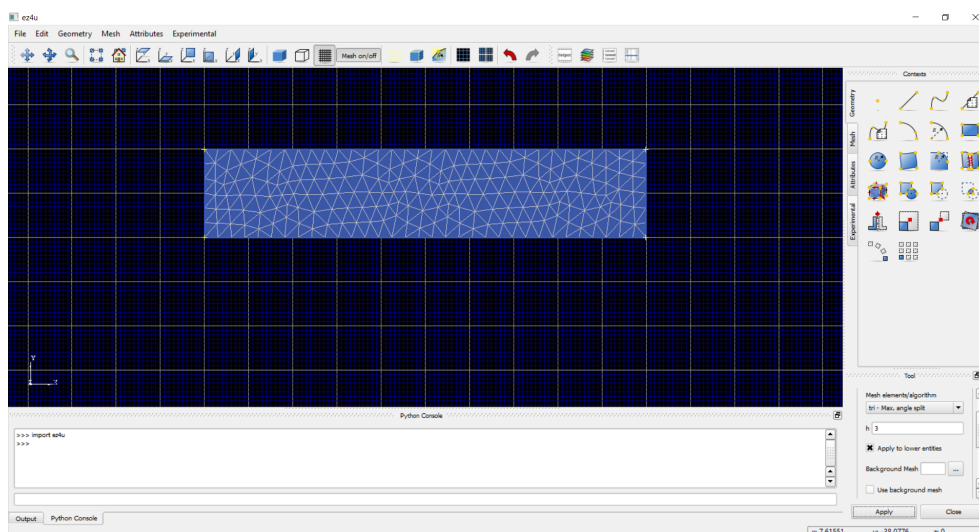


Figura 14: Captura de pantalla del programa EZ4U.

Una funcionalitat molt útil del programa és la capacitat d'incorporar una background mesh. Una background mesh és una malla a la qual a cada node se li assigna un valor numèric. Aquesta s'utilitza per

crear una nova malla, en el mateix domini, que és més fina o més grollera depenent dels valors numèrics assignats als nodes de la malla background. El valor que es dona als nodes és, aproximadament, la mida h dels elements que es formaran al voltant d'aquell node. Per tant, ens interessa assignar valors petits als nodes propers al crack, i valors grans als nodes més llunyans. En l'exemple, escollim com a background mesh una malla triangular amb mida d'element constant $h = 2$ mm (Figura 15). Notem que, com la malla ha de ser simètrica respecte l'eix $y = 0$, només estem considerant la part superior del domini. Posteriorment, s'executa una funció de Matlab que fa la simetria de la malla i les uneix.

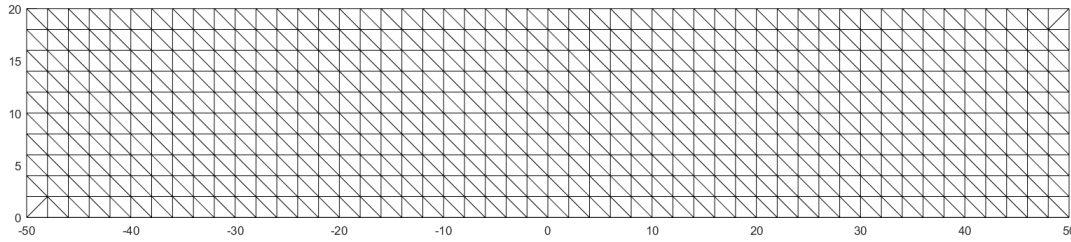


Figura 15: Background mesh.

6.2 Determinació dels valors nodals per la background mesh

El següent pas és determinar els valors nodals de la background mesh. La manera natural de fer-ho és creant una funció $h : \Omega \rightarrow \mathbb{R}$ que depèn de la distància entre el node i el crack. Una forma de senzilla de descriure el crack per poder calcular les distàncies és a través d'una poligonal P . S'escullen suficients punts manualment per anar obtenint la forma que segueix el crack, i entre el segment que formen dos punts consecutius, s'hi afegixen 1000 punts per tenir una discretització suficientment precisa. D'aquesta manera la distància entre un punt qualsevol \mathbf{x} i la poligonal P , i.e, $d(\mathbf{x}, P)$, es pot aproximar com la distància mínima entre el punt \mathbf{x} i els punts que formen la discretització de P . Dit això, definim la funció h com

$$h(\mathbf{x}) = \begin{cases} 0.5 & \text{si } d(\mathbf{x}, P) < 2 \\ \frac{9}{16}d(\mathbf{x}, P) - \frac{5}{8} & \text{si } 2 \leq d(\mathbf{x}, P) < 10 \\ 5 & \text{si } d(\mathbf{x}, P) \geq 10. \end{cases}$$

Així, la nova malla té mida d'element $h = 0.5$ mm a l'entorn del crack, i $h = 5$ mm als punts més allunyats. Amb l'objectiu que el canvi no sigui tan brusc, es fa una transició lineal d' h pels punts intermedis. A la Figura 16 es mostra com es van afegint punts a la poligonal a mesura que anem capturant la trajectòria del crack, i l'evolució respectiva de la malla.

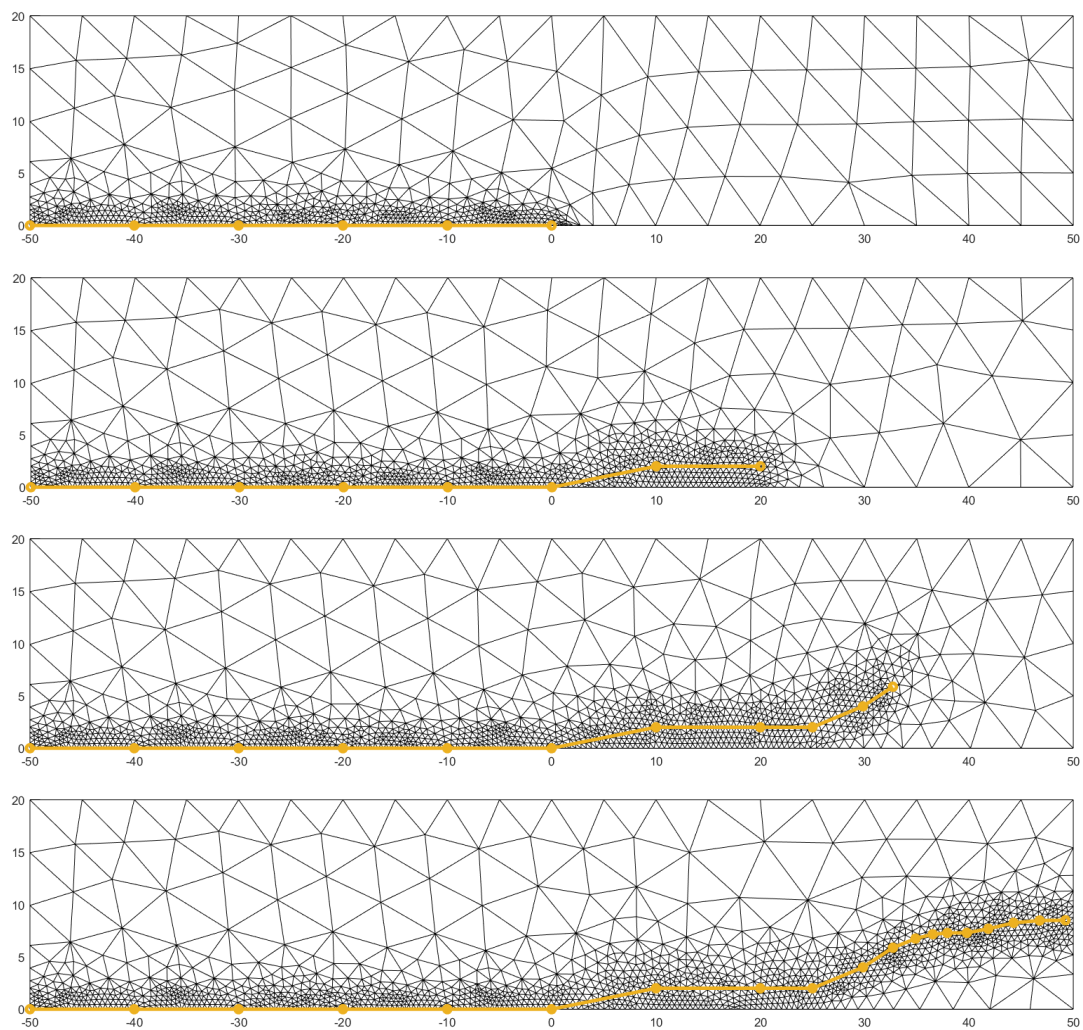


Figura 16: En groc, la poligonal P representant la trajectòria del crack. De fons, la malla resultant.

7. Conclusions

En aquest treball s'ha presentat un model de phase-field dinàmic per a materials de fractura fràgil i se n'ha comprovat l'eficiència en dos problemes clàssics.

El primer problema era de desplaçaments imposats. S'ha considerat una peça en forma de L a la qual se li aplica un desplaçament per provocar-li una fractura. El crack ha sigut capturat perfectament i s'ha interpretat la gràfica de força-desplaçament, que a més, ens ha mostrat convergència respecte al temps.

El segon problema considerat era de crack branching. Consistia en una peça rectangular a la qual se li apliquen forces de tracció molt grans per tal que es creï un crack que pateix una bifurcació. El model ha aconseguit capturar de manera precisa aquesta bifurcació. Posteriorment, s'ha fet una anàlisi energètica del problema i s'ha comprovat la convergència respecte al temps. Addicionalment, hem estudiat quin impacte té la constant material G_c . Hem arribat a la conclusió que a l'augmentar G_c , el crack triga més temps a avançar, però recorre menys espai abans de bifurcar. També hem constatat que si es fa G_c prou petita, el crack no bifurca.

7.1 Coneixements adquirits

En el transcurs d'aquest treball he conegut una àrea que fins ara desconeixia, la de la teoria de la fractura. M'ha semblat un camp molt interessant, en el que he pogut aplicar gran part dels mètodes numèrics explicats en el grau. En particular, he après a utilitzar el mètode d'elements finits per a problemes de mecànica de sòlids i un mètode d'integració temporal molt utilitzat en aquest àmbit com és el de Newmark.

També m'ha servit per prendre consciència de la magnitud d'aquests problemes, que triguen dies a ser resolts per una computadora, i els inconvenients que això comporta. En relació amb això, he hagut d'aprendre a accedir al servidor del LaCàn de forma remota per fer els càlculs necessaris. Finalment, un aspecte que destacaria i que crec que em pot ajudar en un futur, és el fet d'haver après a fer malles i saber modificar-les segons les necessitats.

7.2 Futurs projectes

Durant un treball tan llarg és natural que sorgeixin noves idees que no es poden desenvolupar ja sigui per falta de temps o bé per la dificultat que comporten. Una d'elles i que es va comentar a la secció 6 és la de crear un algorisme que automàticament et modifiqui la malla a mesura que el crack avanci. Una altra que també tindria el seu interès seria la d'implementar un mètode explícit i comparar-lo amb el mètode de Newmark: eficiència, pas de temps necessari per convergir, etc. Per últim, i aquesta és potser la més difícil per tema d'infraestructura, m'agradaria recrear experimentalment els problemes proposats i comprovar si els resultats numèrics obtinguts s'ajusten al que passa a la realitat.

Referències

- [1] Blaise Bourdin, Gilles A Francfort i Jean-Jacques Marigo. "Numerical experiments in revisited brittle fracture". A: *Journal of the Mechanics and Physics of Solids* 48.4 (2000), pàg. 797 - 826.
- [2] Blaise Bourdin, Gilles A Francfort i Jean-Jacques Marigo. "The variational approach to fracture". A: *Journal of elasticity* 91.1-3 (2008), pàg. 5 - 148.
- [3] Rudy JM Geelen et al. "A phase-field formulation for dynamic cohesive fracture". A: *Computer Methods in Applied Mechanics and Engineering* 348 (2019), pàg. 680 - 711.
- [4] Alan Arnold Griffith. "VI. The phenomena of rupture and flow in solids". A: *Philosophical transactions of the royal society of london. Series A, containing papers of a mathematical or physical character* 221.582-593 (1921), pàg. 163 - 198.
- [5] Christian Miehe, Martina Hofacker i Fabian Welschinger. "A phase field model for rate-independent crack propagation: Robust algorithmic implementation based on operator splits". A: *Computer Methods in Applied Mechanics and Engineering* 199.45-48 (2010), pàg. 2765 - 2778.
- [6] Alba Muixí, Sonia Fernández-Méndez i Antonio Rodríguez-Ferran. "A hybridizable discontinuous Galerkin phase-field model for brittle fracture". A: *Reports@ SCM* 4.1 (), pàg. 31 - 42.
- [7] Milton N da Silva Jr, Fernando P Duda i Eliot Fried. "Sharp-crack limit of a phase-field model for brittle fracture". A: *Journal of the Mechanics and Physics of Solids* 61.11 (2013), pàg. 2178 - 2195.
- [8] Thomas Wick. "Modified Newton methods for solving fully monolithic phase-field quasi-static brittle fracture propagation". A: *Computer Methods in Applied Mechanics and Engineering* 325 (2017), pàg. 577 - 611.

A. Forma feble de l'Equació d'Elasticitat Lineal Transitòria

Considerem la forma general de l'equació d'elasticitat lineal transitòria (4) amb condicions de contorn (5). A continuació presentem els passos per deduir la forma feble del problema.

Multipliquem la primera equació de (4) per una funció $\mathbf{v} \in [H^1(\Omega)]^2$ tal que $\mathbf{v} = 0$ a Γ_D i integrem al domini Ω

$$\int_{\Omega} \rho \mathbf{v} \cdot \ddot{\mathbf{u}} \, d\Omega = \int_{\Omega} \mathbf{v} \cdot (\nabla \cdot \boldsymbol{\sigma}) \, d\Omega + \int_{\Omega} \rho \mathbf{v} \cdot \mathbf{b} \, d\Omega.$$

Utilitzant que $\nabla \cdot (\mathbf{v} \cdot \boldsymbol{\sigma}) = \nabla \mathbf{v} : \boldsymbol{\sigma} + \mathbf{v} \cdot (\nabla \cdot \boldsymbol{\sigma})$ i substituint a l'expressió anterior obtenim

$$\int_{\Omega} \rho \mathbf{v} \cdot \ddot{\mathbf{u}} \, d\Omega + \int_{\Omega} \nabla \mathbf{v} : \boldsymbol{\sigma} \, d\Omega = \int_{\Omega} \nabla \cdot (\mathbf{v} \cdot \boldsymbol{\sigma}) \, d\Omega + \int_{\Omega} \rho \mathbf{v} \cdot \mathbf{b} \, d\Omega.$$

Apliquem ara el teorema de la divergència

$$\int_{\Omega} \rho \mathbf{v} \cdot \ddot{\mathbf{u}} \, d\Omega + \int_{\Omega} \nabla \mathbf{v} : \boldsymbol{\sigma} \, d\Omega = \int_{\partial\Omega} \mathbf{v} \cdot \boldsymbol{\sigma} \cdot \mathbf{n} \, d\Gamma + \int_{\Omega} \rho \mathbf{v} \cdot \mathbf{b} \, d\Omega,$$

i usem que $\mathbf{v} = 0$ a Γ_D i les condicions de contorn (5) per obtenir

$$\int_{\Omega} \rho \mathbf{v} \cdot \ddot{\mathbf{u}} \, d\Omega + \int_{\Omega} \nabla \mathbf{v} : \boldsymbol{\sigma} \, d\Omega = \int_{\Gamma_N} \mathbf{v} \cdot \mathbf{t}_N \, d\Gamma_N + \int_{\Omega} \rho \mathbf{v} \cdot \mathbf{b} \, d\Omega.$$

Finalment, utilitzant l'equació constitutiva de (4) i que $\nabla \mathbf{v} = \varepsilon(\mathbf{v})$, resulta l'expressió de la forma feble del problema:

$$\int_{\Omega} \rho \mathbf{v} \cdot \ddot{\mathbf{u}} \, d\Omega + \int_{\Omega} (1-d)^2 \varepsilon(\mathbf{v}) \cdot \mathcal{C} \cdot \varepsilon(\mathbf{u}) \, d\Omega = \int_{\Omega} \rho \mathbf{v} \cdot \mathbf{b} \, d\Omega + \int_{\Gamma_N} \mathbf{v} \cdot \mathbf{t}_N \, d\Gamma_N.$$

B. Codis

B.1 Implementació del model de phase-field dinàmic

```
1 clear all, clc, close all
2 addpath Utilities/
3 addpath referenceElement/
4
5
6 %% Parameters of the problem
7 E = 32; nu = 0.2; rho = 2.45e-9;
8 Gc_scalar = 6e-6; l_scalar = 1.25;
9 t0 = 0; tf = 0.004; nOfSteps = 500; dt = (tf-t0)/nOfSteps;
10 %% Preprocess
11 degree = 1;
12
13 load('meshBranch.mat','-mat');
14 x = X(:,1); y = X(:,2);
15 X2 = [x,-y];
16 T2 = T(:,[1,3,2]);
17 [X,T] = glueMeshes(X,T,X2,T2);
18
19 %Plot malla
20 figure(1),clf
21 kk=plotMesh(X,T);
22
23 nOfElements = size(T,1);
24 nOfElementNodes = size(T,2);
25
26 %Reference element
27 referenceElement = createReferenceElementTri(degree);
28 N = referenceElement.N;
29 %Boundary conditions:
30 x = X(:,1); y = X(:,2); tol=1.e-10;
31 nodesCCD = find(abs(y)<tol); %Nodes on the boundary
32
33 nodesCCDdany = find(abs(y)<0.625 & x <= 0);
34 notCCDdany= setdiff(1:size(X,1),nodesCCDdany);
35
36 XnodesCCD = X(nodesCCD,:); %coordinates of the nodes on the boundary
37 dofCCD = [nodesCCD+size(X,1)];
38
39 TN = connectivityMatrixBoundary(T,1);
40 fN = computeNeumann(X,TN,referenceElement,@normalFluxFunction);
41
42
43 %% Imposition of Dirichlet boundary conditions (system reduction)
44 notCCD= setdiff(1:(2*size(X,1)),dofCCD); %actual degrees of freedom (not boundary
    nodes)
45 Hg = ones(size(N,1),nOfElements); Hg = -Inf*Hg;
46
47 %System solution
48 d = zeros(size(X,1),nOfSteps+1);
49 d(nodesCCDdany,1) = ones(size(nodesCCDdany,1),1);
50 r = zeros(size(dofCCD,1),nOfSteps+1);
```

```

51 u = zeros(2*size(X,1),nOfSteps+1);
52 ut = zeros(2*size(X,1),nOfSteps+1);
53 utt = zeros(2*size(X,1),nOfSteps+1);
54 for i = 1:nOfSteps
55     uCCD = zeros(size(dofCCD,1),1);
56     utCCD = zeros(size(dofCCD,1),1);
57     uttCCD = zeros(size(dofCCD,1),1);
58     [M0,K0,f0] = computeSystemLinearElasticity(X,T,referenceElement,E,nu,rho,d(:,i));
59     f0 = f0 + fN;
60     M = M0(notCCD,notCCD);
61     K = K0(notCCD,notCCD);
62     A0 = rho*M0+dt^2*(1/4)*K0;
63     A = rho*M+dt^2*(1/4)*K;
64     b0 = f0-K0*(u(:,i)+dt*ut(:,i)+(1/4)*dt^2*utt(:,i));
65     b = b0(notCCD) - A0(notCCD,dofCCD)*uttCCD;
66     utt(notCCD,i+1) = A\b;
67     utt(dofCCD,i+1) = uttCCD;
68     u(notCCD,i+1) = u(notCCD,i)+dt*ut(notCCD,i)+dt^2*((1/4)*utt(notCCD,i)+(1/4)*utt(
        notCCD,i+1));
69     u(dofCCD,i+1) = uCCD;
70     ut(notCCD,i+1) = ut(notCCD,i)+dt*((1/2)*utt(notCCD,i)+(1/2)*utt(notCCD,i+1));
71     ut(dofCCD,i+1) = utCCD;
72
73     r(:,i+1) = rho*M0(dofCCD(end),:)*utt(:,i+1) + K0(dofCCD(end),:)*u(:,i+1);
74
75     Gc = Gc_scalar*ones(nOfElements,1);
76     l = l_scalar*ones(nOfElements,1);
77     C = E/((1+nu)*(1-2*nu))*[1-nu, nu, 0; nu, 1-nu, 0; 0, 0, 1-2*nu];
78     [ind_i,ind_j,coef_K,f0,Hg] = FEMSystemDamageField(X,T,referenceElement,Gc,l,Hg,u
        (:,i+1),C);
79     dCCD = ones(size(nodesCCDdany,1),1);
80     Kd0 = sparse(ind_i,ind_j,coef_K);
81     Kd = Kd0(notCCDdany,notCCDdany);
82     f = f0(notCCDdany) - Kd0(notCCDdany,nodesCCDdany)*dCCD;
83     d(notCCDdany,i+1) = Kd\f;
84     d(nodesCCDdany,i+1) = dCCD;
85 end
86
87 %Plots
88 figure(2)
89 plotContinuosSolution(X,T,d(:,end),referenceElement)
90 figure(3)
91 plot(u(1005,2:end), r(end,2:end),'-o')
```

B.2 Càlcul del sistema d'elasticitat lineal

```

1 function [M,K,f]=computeSystemLinearElasticity(X,T,referenceElement,E,nu,rho,d)
2
3 GaussPoints=referenceElement.IPcoordinates;
4 GaussWeights=referenceElement.IPweights;
5 N=referenceElement.N;
6 Nxi=referenceElement.Nxi;
7 Neta=referenceElement.Neta;
8
9 nOfNodes = size(X,1);
10 nOfElements = size(T,1);
```

```

11
12 M=zeros(2*nOfNodes,2*nOfNodes);
13 K=zeros(2*nOfNodes,2*nOfNodes); %global matrix initiallization
14 f=zeros(2*nOfNodes,1);
15 % D=zeros(3,2*nOfNodes);
16
17 %Loop in elements
18 hmin = Inf;
19 for i=1:nOfElements
20     Te=T(i,:); %nodes in the element
21     Xe=X(Te,:); %coordinates of the element nodes
22     de = d(Te);
23     [Me,Ke,fe]=computeElementalMatrices(Xe,GaussPoints,GaussWeights,N,Nxi,Neta,E,nu,
        rho,de,hmin);
24
25     Te2 = [Te, Te+nOfNodes];
26     M(Te2,Te2)=M(Te2,Te2)+Me;
27     K(Te2,Te2)=K(Te2,Te2)+Ke; %assembly
28     f(Te2) = f(Te2) + fe;
29 end
30 hmin;
31
32 end
33 %-----
34 %Computation of elemental matrix and vector
35 function [Me,Ke,fe]=computeElementalMatrices(Xe,GaussWeights,N,Nxi,Neta,E,nu,de)
36
37 nOfNodes = size(Xe,1);
38 Me=zeros(2*nOfNodes);
39 Ke=zeros(2*nOfNodes);
40 Ae = 0;
41 fe=zeros(2*nOfNodes,1);
42 xe = Xe(:,1); ye = Xe(:,2); %x and y coordinates of the element nodes
43
44 C = E/((1+nu)*(1-2*nu))*[1-nu, nu, 0; nu, 1-nu, 0; 0, 0, 1-2*nu];
45
46 %Loop in integration points
47 for k=1:length(GaussWeights)
48     Nk=N(k,:);
49     Nkxi=Nxi(k,:);
50     Nketa=Neta(k,:);
51     dk = Nk*de;
52     %Jacobian of the isoparametric transformation
53     J = [Nkxi*xe Nkxi*ye;Nketa*xe Nketa*ye];
54     %Computation of the derivatives of the shape functions with respect to
55     %x any y
56     Gxy = J\[Nkxi;Nketa];
57     Nkx = Gxy(1,:); Nky = Gxy(2,:);
58     %integration weight at physical element
59     dxy=GaussWeights(k)*det(J);
60     %contribution of the integration point to the elemental matrix
61     B = [Nk, zeros(1, length(Nk));
62         zeros(1,length(Nk)), Nk];
63     D = [Nkx, zeros(1,length(Nk));
64         zeros(1,length(Nk)), Nky;
65         sqrt(2)/2*Nky, sqrt(2)/2*Nkx];
66     Me = Me + B'*B*dxy;

```

```

67     Ke = Ke + ((1-dk)^2)*D'*C*D*dxy;
68 end
69 end

```

B.3 Càlcul del sistema de l'equació de dany

```

1  function [ind_i,ind_j,coef_K,f,Hg]=FEMSystemDamageField(X,T,referenceElement,Gc,l,Hg,
    u,C)
2
3  GaussWeights=referenceElement.IPweights;
4  N=referenceElement.N;
5  Nxi=referenceElement.Nxi;
6  Neta=referenceElement.Neta;
7
8  nOfNodes = size(X,1);
9  nOfElements = size(T,1);
10 nOfElementNodes = size(T,2);
11
12 % K=zeros(nOfNodes,nOfNodes); %global matrix initialization
13 f=zeros(nOfNodes,1);
14 % [mKe,nKe] = size(Ke)
15 mKe = nOfElementNodes;
16 nKe = nOfElementNodes;
17 ind_i = zeros(mKe*nKe,nOfElements);
18 ind_j = zeros(mKe*nKe,nOfElements);
19 coef_K = zeros(mKe*nKe,nOfElements);
20
21 %Loop in elements
22 for i=1:nOfElements
23     Te=T(i,:); %nodes in the element
24     Xe=X(Te,:); %coordinates of the element nodes
25     ue = u([Te,Te + size(X,1)]);
26     Hge=Hg(:,i);
27     [Ke,fe,Hge]=computeElementalMatrices(Xe,GaussWeights,N,Nxi,Neta,Gc(i),l(i),Hge,ue,
        C);
28     Hg(:,i) = Hge;
29
30     coef_K(:,i) = Ke(:);
31     [mi,mj] = meshgrid(Te,Te);
32     ind_i(:,i) = mi(:); ind_j(:,i) = mj(:);
33     f(Te) = f(Te) + fe;
34 end
35
36
37 %-----
38 %Computation of elemental matrix and vector
39 function [Ke,fe,Hge]=computeElementalMatrices(Xe,GaussWeights,N,Nxi,Neta,Gc,l,Hge,ue,
    C)
40
41 J11 = Nxi*Xe(:,1); J12 = Nxi*Xe(:,2);
42 J21 = Neta*Xe(:,1); J22 = Neta*Xe(:,2);
43 detJ = J11.*J22-J12.*J21;
44 %maybe we should use bsxfun instead of diagonal matrices...
45 dvolu = GaussWeights.*detJ;
46 dvolu_diag = diag(dvolu);
47 invJ11 = diag(J22./detJ);

```

```

48 invJ12 = diag(-J12./detJ);
49 invJ21 = diag(-J21./detJ);
50 invJ22 = diag(J11./detJ);
51 % xy-derivatives
52 Nx = invJ11*Nxi + invJ12*Neta;
53 Ny = invJ21*Nxi + invJ22*Neta;
54
55 % Hg = N*Hg;
56 Xg = N*Xe;
57
58 n = size(ue,1);
59 e1 = Nx*ue(1:n/2);
60 e2 = Ny*ue(n/2+1:end);
61 e3 = (sqrt(2)/2)*(Ny*ue(1:n/2) + Nx*ue(n/2+1:end));
62 for i = 1:size(Nx,1)
63     res = (1/2)*[e1(i) e2(i) e3(i)]*C*[e1(i); e2(i); e3(i)];
64     if (res > Hge(i))
65         Hge(i) = res;
66     end
67 end
68
69 fe = 2*N'*(dvolu_diag*Hge);
70 %Elemental matrices
71 Kxxe = Nx'*(dvolu_diag*Nx);
72 Kyye = Ny'*(dvolu_diag*Ny);
73 Ke=Gc*1*(Kxxe+Kyye) + N'*(diag(dvolu.*(Gc/1+2*Hge))*N);

```

B.4 Càlcul de l'energia dissipada

```

1 function E = EnergyDissipation(X,T,referenceElement,Gc,l,d)
2
3 GaussPoints=referenceElement.IPcoordinates;
4 GaussWeights=referenceElement.IPweights;
5 N=referenceElement.N;
6 Nxi=referenceElement.Nxi;
7 Neta=referenceElement.Neta;
8
9 nOfNodes = size(X,1);
10 nOfElements = size(T,1);
11
12 E = 0;
13
14 for i=1:nOfElements
15     Te=T(i,:); %nodes in the element
16     Xe=X(Te,:); %coordinates of the element nodes
17     de = d(Te);
18     Ee = computeEnergy(Xe,Te,de,GaussPoints,GaussWeights,N,Nxi,Neta,Gc,l);
19
20     E = E + Ee;
21 end
22 end
23
24 function Ee = computeEnergy(Xe,de,GaussWeights,N,Nxi,Neta,Gc,l)
25
26 xe = Xe(:,1); ye = Xe(:,2);
27 Ee = 0;

```

```

28
29 for k=1:length(GaussWeights)
30     Nk=N(k,:);
31     Nkxi=Nxi(k,:);
32     Nketa=Neta(k,:);
33     dk = Nk*de;
34     %Jacobian of the isoparametric transformation
35     J = [Nkxi*xe Nkxi*ye;Nketa*xe Nketa*ye];
36     %Computation of the derivatives of the shape functions with respect to
37     %x any y
38     Gxy = J\[Nkxi;Nketa];
39     Nkx = Gxy(1,:); Nky = Gxy(2,:);
40     grad_dk = [Nkx;Nky]*de;
41     %integration weight at physical element
42     dxy=GaussWeights(k)*det(J);
43     %contribution of the integration point to the elemental matrix
44     Ee = Ee + Gc*(dk^2/(2*1) + (1/2)*norm(grad_dk)^2)*dxy;
45 end
46 end

```

B.5 Càlcul de l'energia de deformació

```

1 function E = StrainEnergy(X,T,referenceElement,d,u,C)
2
3 GaussPoints=referenceElement.IPcoordinates;
4 GaussWeights=referenceElement.IPweights;
5 N=referenceElement.N;
6 Nxi=referenceElement.Nxi;
7 Neta=referenceElement.Neta;
8
9 nOfNodes = size(X,1);
10 nOfElements = size(T,1);
11
12 E = 0;
13
14 for i=1:nOfElements
15     Te=T(i,:); %nodes in the element
16     Xe=X(Te,:); %coordinates of the element nodes
17     de = d(Te);
18     ue = u([Te,Te + size(X,1)]);
19     Ee = computeEnergy(Xe,Te,de,ue,C,GaussPoints,GaussWeights,N,Nxi,Neta);
20
21     E = E + Ee;
22 end
23 end
24
25 function Ee = computeEnergy(Xe,de,ue,C,GaussWeights,N,Nxi,Neta)
26
27 xe = Xe(:,1); ye = Xe(:,2);
28 Ee = 0;
29
30 for k=1:length(GaussWeights)
31     Nk=N(k,:);
32     Nkxi=Nxi(k,:);
33     Nketa=Neta(k,:);
34     dk = Nk*de;

```

```

35     %Jacobian of the isoparametric transformation
36     J = [Nkxi*xe Nkxi*ye;Nketa*xe Nketa*ye];
37     %Computation of the derivatives of the shape functions with respect to
38     %x any y
39     Gxy = J\[Nkxi;Nketa];
40     Nkx = Gxy(1,:); Nky = Gxy(2,:);
41     grad_dk = [Nkx;Nky]*de;
42     D = [Nkx, zeros(1,length(Nk));
43          zeros(1,length(Nk)), Nky;
44          sqrt(2)/2*Nky, sqrt(2)/2*Nkx];
45     %integration weight at physical element
46     dxy=GaussWeights(k)*det(J);
47     %contribution of the integration point to the elemental matrix
48     Ee = Ee + ((1-dk)^2)*(1/2)*(D*ue)'*C*D*ue*dxy;
49 end
50 end

```

B.6 Càlcul dels valors nodals de la background mesh

```

1  clear all;
2
3  PrincPoints = [-50,0;-40,0;-30,0;-20,0;-10,0;0,0;10,2;20,2;30,2;40,2;50,2];
4  Points = [];
5  l = linspace(0,1,1000);
6  for i = 1:(size(PrincPoints,1)-2)
7      v = PrincPoints(i+1,:)-PrincPoints(i,:);
8      for j = 1:(size(l,2)-1)
9          points(j,:) = PrincPoints(i,:)+l(j)*v;
10     end
11     Points = [Points; points];
12 end
13 v = PrincPoints(end,:)-PrincPoints(end-1,:);
14 for j = 1:size(l,2)
15     points(j,:) = PrincPoints(end-1,:)+l(j)*v;
16 end
17 Points = [Points; points];
18
19 load('meshBranchBG.mat','-mat');
20 ponderacions = zeros(size(X,1),1);
21 for i = 1:size(X,1)
22     dist = Inf;
23     for j = 1:size(Points,1)
24         if (norm(X(i,:)-Points(j,:)) < dist)
25             dist = norm(X(i,:)-Points(j,:));
26         end
27     end
28     param = 0.5;
29     if (dist <= 2) ponderacions(i) = param;
30     elseif (dist <= 10) ponderacions(i) = ((5-param)/8)*dist + 5 - 10*((5-param)/8);
31     else ponderacions(i) = 5;
32     end
33 end
34 num = 1:size(X,1);
35 num = num';
36 ponderacions = [num,X,ponderacions];
37 num = 1:size(T,1);

```

```
38 num = num';  
39 elements = [num,T];
```